A Character-level Ngram-based MT Approach for Lexical Normalization in Social Media

Anonymous ACL submission

Abstract

This paper presents an ngram-based MT approach that operates at character-level to generate possible canonical forms for lexical variants in social media text. It utilizes a joint n-gram model to learn edit sequences of word pairs, thus overcomes the shortage of phrase-based approach that is unable to capture dependencies across phrases. We evaluate our approach on two English tweet datasets and observe that the ngram-based approach significantly outperforms phrase-based approach in normalization task. Our simple model achieves a broad coverage on diverse variants which is comparable to complex hybrid systems.

1 Introduction

As large volume of text data being produced daily on social media platforms, user-generated text has become the biggest source for text mining and Natural Language Processing (NLP) tasks. It contains vast valuable information and reflects users' diversified habits of readings and writings, which had been known to be deviated from standard language usage (Eisenstein, 2013). Many words are written in non-standard forms, either being diversely abbreviated, respelled, or mistyped. This can lead to explosive growth of vocabulary size for any NLP models, aggravated unknown token and data sparseness problems. Double-digit performance decreases had been widely observed in basic NLP tasks, such as part-of-speech tagging (Gimpel et al., 2011), named entity recognition (Ritter et al., 2011), and parsing (Foster et al., 2011). Additionally, it also causes communication gap between users that across diverse communities. Users who are not familiar with certain language habits, e.g. domain-specific abbreviations, or regional accents that are explicitly written, may find the text difficult to understand.

Lexical normalization is a task that aims to establish correlations between standard words and their diverse written forms in user-generated text, so that text audiences (either users or machines) can be informed of possible standard forms of unknown tokens. Normalization models have either served as preprocessing tools for downstream tasks (Han et al., 2013), or necessary components in joint decoding models (Li and Liu, 2015). It is an important task for developing more robust NLP applications on social media.

The connections between standard words and lexical variants have been generally represented as a noisy channel model $\arg\max P(T|S) = \arg\max P(S|T)P(T)$ that aims to find the most probable target sequence $\hat{T}$ for given source sequence S. One major challenge is that it is difficult to estimate reliable $P(S|T)$ because the training data is limited. Due to the productive and creative natures of social media language, it is impractical to collect sufficient data to cover all spelling variants for standard words.

Two kinds of solutions have been proposed to approximate $P(S|T)$. The similarity-based approaches used Edit Distance (ED), Longest Common Subsequence (LCS) or any combination of the two to measure lexical similarities between letter and phone sequences, while incorporating different context-based models to capture distributional similarities in unlabeled corpora (Han and Baldwin, 2011; Hassan and Menezes, 2013; Yang and Eisenstein, 2013). The problem is that similarity-based metrics are obviously restricted to specific variants that are orthographically or phonetically close to standard forms. Another solution views $P(S|T) = \prod_i P(s_i|t_i)$ as a string transduction subtask at subword-level. The mapping from word t_i to s_i has been represented as edits of characters or character-blocks, and the weights of edits were estimated through diverse models, such as edit-probability model (Cook and Stevenson, 2009), weighted rewrite model (Beaufort et al., 2010), character-level phrase-based Machine Translation

(MT) model (Pennell and Liu, 2011) and sequence labeling model, specifically Conditional Random Field (CRF) (Liu et al., 2011; Chrupała, 2014). The MT framework has the advantage that it is very flexible to handle diversified and productive variation types such as arbitrary letter transpositions or repetitious typing. But the phrase-based translation model can not capture the dependencies across phrases, that makes it unable to deal with cases like highly abbreviated words. In another aspect, the sequence labeling framework is good at capturing dependencies between edits. But it is less flexible than MT framework because it operates in monotonous manner and requires the length $|t_i|$ to be strictly equal to $|s_i|$.

In this paper, we propose a more general string transduction model that learns edit sequences of word pairs with a joint n-gram model under MT framework. Our model overcomes the shortcomings of both phrase-based translation model and sequence labeling framework, and can flexibly handle words with small variations or highly abbreviated words. Since recovering highly abbreviated words requires to infer arbitrary long consecutive insertions for input strings, we propose to transform them into Finite State Automata (FSA) with loops of insertion points between adjacent letters. The whole inference process is efficiently implemented through the standard operations of the Weighted Finite-State Transducer (WFST) framework, OpenFST (Allauzen et al., 2007).

The main contributions of our work are: (i) we show that an ngram-based MT approach significantly outperforms phrase-based approach in the context of lexical normalization. Our simple model can achieve a broad coverage on diverse variants which is comparable to complex hybrid systems; (ii) we investigate how string transduction models perform in a more general normalization task that includes phrasal abbreviations as targets.

2 Methods

Given a source string $S = (s_1, s_2...s_n)$, the goal of normalization task is to find a target string $T = (t_1, t_2...t_n)$ that maximizes $P(T|S)$. It can be factored as a noisy channel model: $\arg\max_T P(T|S) = \arg\max_T P(S|T)P(T)$. In the context of lexical normalization, $P(S|T) = \prod_i P(s_i|t_i)$ is a lexicalized transformation model that generates possible standard words $\{t_i\}$ for each variant token s_i, and $P(T)$ is a Language Model (LM) that select the most probable sequence $\hat{T}$. In this work, we focus on learning the lexicalized transformation model $P(s|t)$.

2.1 Model

Due to the limited training pairs, it is difficult to directly estimate reliable $P(s|t)$. Pennell and Liu (2011) has proposed to estimate the posterior probability $\arg\max_t P(t|s) = \arg\max_t P(s|t)P(t)$ that leverages a character LM $P(t)$ to learn standard word formations from unlabeled data and ensure that most generated candidates are reasonable words. Based on this framework, we suggest an alternative model to estimate $P(s|t)$ instead of phrase-based translation model.

The proposed model is based on a joint n-gram model, which has long been used as an alternative to phrase-based model in the context of machine translation (Crego et al., 2005). Our model differs in that it operates at character-level. A pair of letter sequences (s, t) is represented as a sequence of edit operations $E = (e_1, ...e_J)$, so that the joint probability $P(s, t)$ is estimated as:

$$P(s,t) \approx \prod_{j=1}^{J} P(e_j|e_{j-n+1}...e_{j-1})$$

The conditional probability is then computed as follows:

$P(s|t) = \frac{P(s,t)}{P_{marg}(t)}$

where $P_{marg}(t) = \sum_s P(s,t)$ is the marginal probability of t in the joint n-gram model.

The lexicalized transformation model is defined as:

$\hat{t} = \arg\max_t \frac{P(s,t)}{P_{marg}(t)} P_{LM}(t)D(\alpha)L(t)^{w_l}$

where $D(\alpha) = \alpha^2$ is a simplified distortion model to handle transpositions of adjacent letters, and $L(t) = e^{length(t)}$ is a length penalty to control the length of target sequence, which is similar to word penalty in MT. These two terms turn out to be very useful in normalization task.

2.2 WFST-based Training and Decoding

We adopt the training scheme proposed by Novak et al. (2012) to estimate the joint n-gram model[1]. The pairs of letter sequences are first aligned into edit sequences through a stochastic transducer which is optimized by Expectation-Maximization (EM) algorithm (Ristad and Yianilos, 1998). The edit sequences are used to estimate a standard n-gram model, which is then transformed into a

[1] http://code.google.com/p/phonetisaurus (BSD-3-Clause)

WFST with source input and target output labels. Another training method can also be used to estimate joint n-gram model (Bisani and Ney, 2008), but it is slower than the method we used.

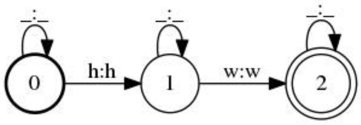

Figure 1: An example FSA of input string `hw`. The symbol _ denotes insertion points.

Decoding process takes a cyclic FSA as input, as shown in Figure 1. We first compose the input FSA I with the joint N-gram model M to produce a mapping from input string to all possible candidates. Since it is a cyclic WFST that encodes unbounded candidate strings, we need to prune it into acyclic WFST with $ShortestPath$ operation, that preserves global K-shortest paths. The process is described as follows:

$C = Det(Proj_o(ShortestPath(I \circ M)))$

where $Proj_o$ projects output labels of the WFST into a FSA that encodes only candidate strings, and Det determinizes the candidate FSA.

To compute the marginal probabilities of candidates, we take the unweighted candidate lattice C_{uw} as input. It is composed with M to generate all possible variant strings for the candidates. The weights of all the paths that generate the same candidate are added together through Det operation, as described below:

$C_{marg} = Det(Proj_o(M \circ C_{uw}))$

Hence, we get the final top-m candidate list C_{list} through the following computation:

$C_{list} = ShortestPath(C \circ [C_{marg}]^{-1} \circ LM_{ch})$

3 Experiments

We concentrate on evaluating the proposed model in word-level normalization, which only generate candidates for given tokens but does not rerank them according to surrounding contexts. We do not consider sentence-level normalization in this work because it requires to collect an extra corpus to estimate word-level LM in target domain. One issue is that there is no consensus on what kind of data represents the target domain, as argued by Chrupała (2014). As a variety of corpora have been used to estimate LM (Liu et al., 2012; Xu et al., 2015; van der Goot and van Noord, 2017), what data constitutes the target domain is still unknown.

Experimental settings

The proposed model is evaluated under two different settings on English tweet datasets, which have been anonymized and potentially contain offensive words. Offensive words are usually actively respelled in tweets to avoid detection, thus they are main targets in normalization task.

(1) A traditional setting is to normalize only single-token words, that excludes phrasal abbreviations like (`imo`, `in my opinion`). To make the results comparable to previous work, we follow the experimental setup in Li and Liu (2014), that used 2,333 unique word pairs in the annotated data (Pennell and Liu, 2011) (GPL-3.0) for training, and 569 unique word pairs in the **Lexnorm** dataset (Han and Baldwin, 2011) (CC-BY 3.0) for testing.

Under this setting, we use the CMU dictionary[2] to define the scope of target words, and collect about one million English tweets as background corpus in which each tweet contains at least three in-vocabulary (IV) words. All IV words in the corpus are extracted to estimate the character LM.

(2) A more general setting has been proposed by the shared task of the 2015 Workshop on Noisy User-generated Text (**W-NUT**) (Baldwin et al., 2015), that includes phrasal abbreviations as targets of normalization. It has suggested the systems to be evaluated under two modes: the constrained mode asks the systems to use training data as the only information source, while the unconstrained mode allow the systems to utilize any accessible textual resource. In the W-NUT dataset, there are 1,183 unique word pairs in training data and 907 unique word pairs in testing data.

Under this setting, we only evaluate the proposed model under constrained mode because it is difficult to capture phrasal abbreviations in open background corpus. The dictionary that defines target words or phrases is compiled from training data that contains 2,950 annotated tweets. All IV words/phrases in training data are used to estimate the character LM, in which the spaces in phrases are replaced with `<space>` symbol.

Baselines

We compare our Ngram-Based MT approach (**NBMT**) with three baselines, including a simple dictionary-based approach that generates candidates directly from training word pairs (**Dict**), an ngram-based approach that generates candidates base on the joint probability $P(s,t)$ (**Joint**), and

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

	Accuracy %				
Method	Top1	Top3	Top10	Top20	Cover
Dict					36.4
Joint	42.0	55.4	66.8	72.2	84.9
PBMT	67.3	76.4	79.3	79.8	81.0
NBMT	**69.6**	**81.0**	**89.6**	**91.6**	**93.8**
Sys1	73	81.9	86.7	89.2	94.2
Sys2	77.1	87.0	93.0	94.8	95.9

Table 1: N-best accuracy on Lexnorm dataset. Sys1 is a hybrid system which contains 4 subsystems, including a spell checker, two MT systems and one sequence labeling system (Li and Liu, 2012); Sys2 reranks the candidates generated from 5 subsystems with a maximum entropy model (Li and Liu, 2014).

	LD <= 2 (479 pairs)		LD >2 (90 pairs)	
Method	Top3	Cover	Top3	Cover
PBMT	82.0	87.1	46.7	48.8
NBMT	85.4	97.3	57.8	75.6

Table 2: Performances comparison on grouped pairs in Lexnorm dataset. LD denotes the length differences between source and target strings.

the character-level Phrase-Based MT approach implemented by Moses (Koehn et al., 2007) (**PBMT**). A 5-gram character LM is used in both PBMT and NBMT. The order of joint n-gram model is set to 3. All approaches generate at most 200 candidates for each token in test set.

3.1 Evaluation on Lexnorm Dataset

Table 1 shows the results of our experiment on the Lexnorm dataset. The result of dictionary-based method tells how many pairs in test set are covered by training pairs. The simple Joint approach achieves better coverage than PBMT, which indicates the superiority of joint ngram model over phrase-based model in normalization task. Our proposed model NBMT significantly outperforms PBMT, and achieves the coverage that is close to the complex hybrid systems.

Table 2 shows the performance comparison between NBMT and PBMT on pairs grouped by length difference. In the test set, most pairs (84%) have length differences that are less or equal to 2, and they are handled well by both approaches. The pairs with LD >2 (16%) are the difficult part in normalization task. Our NBMT approach performs much better than PBMT on these pairs.

	Accuracy %				
Method	Top1	Top3	Top10	Top20	Cover
Dict					42.1
Joint	33.2	47.1	57.7	63.1	75.9
PBMT	52.7	62.8	66.4	67.4	70.3
NBMT	**52.7**	**67.1**	**74.6**	**77.6**	**83.0**

Table 3: N-best accuracy on W-NUT dataset.

	LD <= 2 (666 pairs)		LD >2 (241 pairs)	
Method	Top3	Cover	Top3	Cover
PBMT	71.9	81.4	37.8	39.8
NBMT	75.7	93.1	43.6	55.2

Table 4: Performances comparison on grouped pairs in W-NUT dataset. LD denotes the length differences between source and target strings.

3.2 Evaluation on W-NUT Dataset

Table 3 shows the results of our experiment on the W-NUT dataset. Significant performance drops of all approaches are observed due to two factors: first, the inclusion of phrasal abbreviations introduces extra ambiguities when normalizing given tokens; second, the character LM used in PBMT and NBMT is estimated by limited training corpus. Overall, the NBMT approach consistently outperforms PBMT, except that they achieve equal top-1 accuracy.

In Table 4, we can see that the pairs with LD >2 account for over 26% of total pairs in test set. This indicates that normalization task on W-NUT dataset is more difficult than on Lexnorm dataset. The NBMT approach still outperforms PBMT on pairs (LD >2) but the gap is smaller than Table 2.

4 Conclusion

MT framework is more suitable for normalization task than sequence labeling framework because it possesses the flexibility to handle arbitrary variation types. Our ngram-based MT approach effectively overcomes the shortage of phrase-based approach and achieves result that is comparable to complex hybrid systems in word-to-word normalization. However, there is still a lot of room for improvement when considering word-to-phrase normalization. We are planning to tackle this problem in the future.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23, Berlin, Heidelberg. Springer Berlin Heidelberg.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China. Association for Computational Linguistics.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing SMS messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 770–779, Uppsala, Sweden. Association for Computational Linguistics.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Commun.*, 50(5):434–451.

Grzegorz Chrupała. 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–686, Baltimore, Maryland. Association for Computational Linguistics.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78, Boulder, Colorado. Association for Computational Linguistics.

Josep M. Crego, Marta R. Costa-Jussa, Jose B. Marino, and Jose A. R. Fonollosa. 2005. Ngram-based versus phrase-based statistical machine translation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 893–901, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, Portland, Oregon, USA. Association for Computational Linguistics.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.*, 4(1).

Hany Hassan and Arul Menezes. 2013. Social text normalization using contextual graph random walks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1577–1586, Sofia, Bulgaria. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Chen Li and Yang Liu. 2012. Improving text normalization using character-blocks based models and system combination. In *Proceedings of COLING 2012*, pages 1587–1602, Mumbai, India. The COLING 2012 Organizing Committee.

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 86–93, Baltimore, Maryland, USA. Association for Computational Linguistics.

Chen Li and Yang Liu. 2015. Joint pos tagging and text normalization for informal text. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1035–1044, Jeju Island, Korea. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011. Insertion, deletion, or substitution? normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 71–76, Portland, Oregon, USA. Association for Computational Linguistics.

Josef Novak, Paul Dixon, Nobuaki Minematsu, Keikichi Hirose, Chiori Hori, and Hideki Kashioka. 2012. Improving wfst-based g2p conversion with alignment constraints and rnnlm n-best rescoring. In *INTERSPEECH*, volume 3.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of SMS abbreviations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 974–982, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *Computational Linguistics in the Netherlands Journal*, 7:129–144.

Ke Xu, Yunqing Xia, and Chin-Hui Lee. 2015. Tweet normalization with syllables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 920–928, Beijing, China. Association for Computational Linguistics.

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 61–72, Seattle, Washington, USA. Association for Computational Linguistics.